DUAL PROCESSOR TRUSTED COMPUTING ENVIRONMENT

FIELD OF THE INVENTION

[0001] The present invention relates to secured multi-application programming on a dual component electronic computing device.

BACKGROUND OF THE INVENTION

[0002] Prior art is described in applicant's 5,513,133, 5,742,530, WO 98/50851, WO99/26184 and WO 00/42484.Other prior art is described in US Patent 4,742,215, and in WO 00/48416, and WO 99/01848 and in an article by Pierre Girard and Jean-Louis Lanet, from Gemplus International, "New Security Issues Raised by Open Cards", Information Security Technical Report, vol. 4, pp. 19-27, Elsevier, May 1999 hereinafter Girard.

SUMMARY OF THE INVENTION

[0003] Computing platforms intended for electronic commerce or for storing and processing data for other applications are constantly being attacked by competitors, vandals, and other inimical entities. The problem of devising secure systems becomes more intricate as new and more flexibile and complex modes of operation emerge. In the realm of ubiquitous computing, typically smart cards and subscriber identification modules within telephone handsets are expected to allow the execution of programs originated by a plurality of entities, and, as described in the article by Girard et.al. supra even to allow the loading of new application programs when the system is in service in the field. Another emerging need is to combine processing and security features with the capability of storing significant amounts of data on small size devices such as the device disclosed in the applicant's United States Patent 5,519,843. While solutions have been proposed for insuring security in such platforms, as described in the section about the background of the invention supra, the extent of security insured in actual fact is largely unknown, because attacks and methods for breaking security arrangements are not always foreseen. The prevalent usage of a single processing engine which serves both for executing application programs that cannot enjoy the same level of trust as the inner

core of the system as well as for executing the operations belonging to the inner core, as done in current art designs, in principle decreases the level of confidence about the security of the system. More confidence about the security of the system can be gained by defining logical borders between components and sub-modules of the system and by defining and adhering to formal rules governing the interactions between the components and sub-modules. The current invention shows how separation and logic borders between components and sub-modules of the system can be defined and embodied in a material realization, in association with security rules, where one of the methods being adopted to accomplish the separation being the usage of at least two processing engines instead of a single processing engine.

[0004] It is the purpose of the current invention to provide a secure computation system achieving the same level of security accomplished by closed, rigid, more rudimentary, security application modules (SAMs) while providing an open and flexible environment, operable to serve multiple applications associated with multiple agents and with remote downloads in the field. Further objects of the invention are to provide a system operable to control secured repositories of data and programs, to support protected mutually exclusive execution of programs, to control the operation and to store the results of electronic value applications and transactions, to enable authorized downloading of data for storage, to enable authorized downloading of programs for execution. A further object of the invention is to provide a design for a secure data module implementable on a small mobile devices. A further object of the invention is to provide a design for a secure data module containing "of the shelf" prior art core of a more rudimentary security application module (SAM), integrated with circuitry serving for implementing the complementary part of the system. A further object of the invention is to create a secure data module where symmetric key and public key cryptographic techniques work in a closed tamper reistant environment. A further object of the invention is to create a secure data module integrated with physical devices for confidentialy authenticating a user, typically fingerprint and PIN (Personal Identification Number) readers. A further object of the invention is to create a secure data module compliant with existing and forthcoming standardization, as described in the section "background of the invention" supra. In one typical implementation, the system is imbedded within a wireless communication device. In a further typical use,

the system is operable to support credit or debit charge card public key protected clearance scheme. In a further typical use, the system serves as a mobile agent for airlines including of an updatable repository of air flight schedules, automatic airline reservation and ticketing scheme with payment implemented using a PKI charge clearance network.

[0005] In a preferred embodiment of the invention, the system includes

a first component operable to insure authorized access to the secured repositories of data and programs, to manage and control the secured repositories of data and programs, to insure the integrity of the secured repositories of data and programs, and to prevent one applicatio from utilizing, scrutinizing or modifying another application, and

a second component operable to execute authorized applications, the first and second components operating in parallel.

[0006] In another preferred embodiment, portions of a complete program are being fetched during run time from the repositories of programs with access to locations within the repositories of programs not allowed for the program being blocked by a firewall, the firewall including a mask of control values corresponding to a plurality of segments within the repositories of programs and controlled by the first component.

[0007] In a preferred embodiment of the invention, the system is imbeded within a single monolithic microelectronic integrated circuit. In another preferred embodiment, traffic of information between parts of the system that do not reside on a common monolitic IC chip is encrypted.

[0008] In a preferred embodiment of the invention, the internal circuitry is protected by physical tamper resist methods and containing D/A (Digital to Analog) converters, operable to hide data contents, typically audio signals in a digital format within the data repositories while exporting the contents in an analog form only.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Fig. 1 is a generic simplified overview block representation of preferred embodiments of a structure for a dual processing trusted computing environment (TCE);

Fig. 2 is a simplified block diagram of an architecture of a secure data module serving as a trusted computing environment, showing a formal division of the system into sub-modules and their interactions;

Fig. 3 is a simplified block diagram of a prior art architecture of a securable public key protected device with a closed operating environment;

Fig. 4 is a block diagram of a double processor Public Key protected device with an appended protected environment operable to execute programs downloaded from authorized application issuers wherein issuer applications are typically stored in NAND Flash ROM;

Fig. 5 is a block diagram of a double processor Public Key protected device with an appended protected environment operable to execute programs downloaded from authorized application issuers wherein issuer applications are typically stored in NOR Flash ROM;

Fig. 6 is an extended file structure with a memory storage extension derived from a mature prior art closed Public Key Protected EEPROM application file structure;

Fig. 7 is a preferred embodiment of an access control enable vector for masking the access by an application fetched from NOR Flash memory;

Fig. 8 is a flow chart demonstrating the process of authentication and of regulated file access;

Fig. 9 is a block diagram of a multi-media synchronized application operable to output a plurality of synchronized analog signals; and

Fig. 10 is a block diagram of a trusted computing environment (TCE) operable to process biometric inputs for user authentication.

DETAILED DESCRIPTION OF THE INVENTION

Preferred Formal Embodiments

[0010] The computing system illustrated in Fig. 1 is a simplified block diagram of a trusted computing environment (TCE) 1, interfaced to a Host, 4. Said host, preferably endowed with Public Key Infrastructure (PKI), can be a simple or composite computing system operable to authorize, after positive PKI identification, transactions, data transfers, to an from a protected data repository and to authorize execution of

applications in the Trusted Application Computing Environment, (TACE) 3, originated from application issuers recognized by the TCE Security Application Module (SAM), 2.

[0011] The SAM is preferably a PKI protected module, operable to perform sensitive transactions, to negotiate and approve access to a PKI authorized application, and to convey access to the particular application data prescribed in the PKI negotiation between the TCE SAM and the Host. Access is granted to a complete or partial portion of the application according to the priority and conditions, which are provided in the Host PKI Certificate, issued by the Application Certification Authority.

[0012] The PKI Master typically communicates and mutually authenticates conditions of participation in an application with the Host, prior to executing an application. Files for applications which are computed in the TACE, 3, are either downloaded by the TCE SAM from a larger memory repository, typically from a NAND type Flash memory, typically into executable volatile RAM or in another preferred embodiment based on NOR type Flash memory, authorized blocks of the address enable are unmasked for the specific application for direct CPU execution.

[0013] During the process of the application session, the computing environment is authorized to conduct financial negotiations with the TCE SAM, virtually as a trusted smart card reader would negotiate with a PKI smart card's SAM. The transactions between the TACE and the SAM are typically executed in ISO 7816 type formats through the security I/O buffer, 5.

[0014] Typically, all downloads of Application Files, are PKI negotiated and written into memory exclusively by the TCE SAM. Typically, electronic commerce and purse transactions are executed by the TCE SAM and all changes in non-volatile purses are written in non-volatile memory by the TCE SAM.

[0015] In a preferred embodiment, the TACE is operative to execute biometric and imaging computations, collecting data via an analog to digital interface. Results of large scale computations are typically output in analog mode operable from a digital to analog converter, D/A, and in digital form to the Host and/or to the TCE SAM.

[0016] The computing system illustrated in Fig. 1 is a simplified block diagram a trusted computing environment, (TCE) 1, interfaced to a Host, 4. A Public Key Infrastructure (PKI) Host can be a simple or composite computing system operable to authorize, after positive PKI identification, transactions, data transfers, to an from a

defined protected application directory and to authorize execution of applications in the Trusted Application Computing Environment, (TACE) 3, from an application issuer recognized by the TCE Security Application Module (SAM), 2.

[0017] In Fig. 2 we describe an architecture of a TCE (Trusted Computing Environment) secure data module that enhances the capabilities of a SAM cryptocomputer core by appending an environment for application execution. The aim is to preserve the same level of security that is attained by the closed, typically rudimentary, SAM core has limited accommodation for loadable applications, and in addition to ensure hardware separation between different applications. To achieve this aim we propose methods by which

1) the plurality of functions of the TCE system are identified at formal levels, and the system is divided accordingly into sub-modules;

2) the plurality of interactions between the sub-modules are defined at the formal level, and security assumptions of these interactions are set;

3) the design is translated into an integrated circuit embodiment, using computer simulations, in a mode that ensures that assumptions on the interaction between sub-modules, as defined in the more formal level, are implemented in a physical embodiment.

[0018] This formalization produces a design, which typically satisfies a concise set of security-related assumptions. The activation of this method generates an architecture that comprises

1) sub-modules that typically are operable to fulfill the function of a closed, rudimentary SAM core, and

2) sub-modules that typically are operable to accommodate loadable applications,

such that each of the above two groups of sub-modules typically constitutes an autonomous computing environment, including an autonomous processing engine. A suitable design in the formal level includes the sub-modules of Fig. 2, shown with their typical interactions. A description of the plurality of functions of these sub-modules, and of the interactions between sub-modules including the security-related assumptions follows. In the description we distinguish between a "data interaction", which is represented by a full line in Fig. 2, and a "control interaction", which is represented by a dashed line. The sub-modules enclosed within

the dashed box 570 in Fig. 2 are typically appended to the SAM core to accommodate the loadable applications.

I/O APPARATUS (500):

[0019] Interfaces between the secure data module and the external computational environment, typically a host terminal, a telephone handset, a TV set top box, or a smart card reader. The data traffic that flows via this sub-module may include commands and responses exchanged between the external computational environment and the management and control apparatus (510); data exchanged between the external computational environment and the data repository (530); and loadable applications transferred from the external computational environment to the EXECUTABLES REPOSITORY (550).

MANAGEMENT & CONTROL APPARATUS (510):

[0020] Manages and controls the plurality of resources in the system, negotiates with the external computational environment, and communicates with physical environment devices, typically a fingerprint sensor or another user authentication device. In a preferred embodiment, all or part of the functionality of the management and control apparatus is typically implemented by the processing engine of the SAM core, and the complementary part is implemented by the processing engine of the appended environment for loadable applications.

APPARATUS FOR AUXILIARY COMPUTATIONS (520):

[0021] Performs specialized computations, such as those needed for cryptographic operations.

DATA REPOSITORY (530):

[0022] Stores data on a non-volatile medium, typically operable to be accessed and modified directly or indirectly by authorized commands and applications.

COORDINATION MEDIUM (540):

[0023] Is typically operable to buffer between the data repository (530), and the external computational environment and execution stage (560).

EXECUTABLES REPOSITORY (550):

[0024] Is typically operable to retain authorized executables on a non-volatile medium.

EXECUTION STAGE (560):

[0025] Is operable to provide an infrastructure for executing an application. In a preferred embodiment, this infrastructure typically includes a processing engine, a random access memory for code and data, and system software that typically includes high or intermediate level language interpreters.

[0026] Data interaction between the EXTERNAL COMPUTATIONAL ENVIRONMENT and the I/O APPARATUS:

Role: To transfer commands, responses, data, and executables.

[0027] Data interaction between the I/O APPARATUS and the MANAGEMENT & CONTROL APPARATUS:

Role: To transfer commands and responses.

[0028] Data interaction between the I/O APPARATUS and the COORDINATION MEDIUM:

Role: To transfer data.

[0029] Security-related assumptions:

When this interaction is operating, the interactions between the coordination medium and the data repository and execution stage are detached.

[00030] Data interaction between the I/O APPARATUS and the EXECUTABLES REPOSITORY:

Role: To transfer executables.

[0031] Data interaction between the MANAGEMENT & CONTROL APPARATUS and the PHYSICAL ENVIRONMENT DEVICES:

Role: Typically to transfer an authentication of the person operating the secure data module.

[0032] Data interaction between the MANAGEMENT & CONTROL APPARATUS and the APPARATUS FOR AUXILIARY COMPUTATIONS:

Role: Typically to transfer operands and results of cryptographic and arithmetic computations.

[0033] Security-related assumptions:

When this interaction is operating, the interaction between the apparatus for auxiliary computations and the execution stage is detached.

[0034] Data interaction between the MANAGEMENT & CONTROL APPARATUS and the DATA REPOSITORY:

Role: To transfer data.

[0035] Security-related assumptions:

When this interaction is operating, the interaction between the coordination medium and the data repository is detached.

[0036] Data interaction between the APPARATUS FOR AUXILIARY COMPUTATIONS and the DATA REPOSITORY:

Role: To transfer bulk data, typically files.

[0037] Security-related assumptions:

When this interaction is operating, the interaction between the coordination medium and the data depository is detached.

[0038] Data interaction between the APPARATUS FOR AUXILIARY COMPUTATIONS and the COORDINATION MEDIUM:

[0039] Data interaction between the APPARATUS FOR AUXILIARY COMPUTATIONS and the EXECUTION STAGE:

Role: Typically to transfer operands and results of cryptographic and arithmetic computations.

[0040] Security-related assumptions:

When this interaction is operating, the interactions between the apparatus for auxiliary computations and the management and control apparatus and the data depository are detached.

[0041] Data interaction between the DATA REPOSITORY and the COORDINATION MEDIUM:

Role: To transfer data.

[0042] Security-related assumptions:

When this interaction is operating, the interactions between the coordination medium and the I/O apparatus and the execution stage are detached.

[0043] Data interaction between the COORDINATION MEDIUM and the EXECUTION STAGE:

Role: To transfer data and authorized requests to access data.

[0044] Security-related assumptions:

When this interaction is operating, the interactions between the coordination medium and the I/O system and the data repository are detached.

[0045] Data interaction between the EXECUTABLES REPOSITORY and the EXECUTION STAGE:

Role: To transfer loadable executables.

[0046] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the I/O APPARATUS:

Role: To select the type of data interaction in which the I/O apparatus is engaged.

[0047] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the APPARATUS FOR AUXILIARY COMPUTATIONS:

Role: To select the types of computations in which the apparatus for auxiliary computations is engaged, and to select in a typically dynamic manner the sub-modules to which the apparatus for auxiliary computations is attached.

[0048] Security-related assumptions:

The data interactions between the apparatus for auxiliary computations and all the sub-modules except of the execution stage are detached whenever the apparatus for auxiliary computations is attached to the execution stage.

[0049] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the DATA REPOSITORY:

Role: To select the type of operation in which the data repository is engaged, to scrutinize the coordination medium, and to govern the attachments and detachments of the coordination medium to and from the data repository.

[0050] Security-related assumptions:

An attachment between the coordination medium and the data repository is allowed only after the coordination medium is detached from the I/O apparatus and from the execution stage, and its contents scrutinized as to verify that these contents constitute a proper and authorized command.

[0051] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the COORDINATION MEDIUM:

Role: to select the sub-module to which the coordination medium is attached, and to scrutiny the contents of the coordination medium when the coordination medium contains a request to access the data repository so as to verify that the request is authorized.

[0052] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the EXECUTABLES REPOSITORY:

Role: To control the import of executables from the external computational environment into the executables repository.

[0053] Control interaction between the MANAGEMENT & CONTROL APPARATUS and the EXECUTION STAGE:

Role: To control the loading of executables from the executables repository into the execution stage, to set the mask serving as a firewall in the fetching of portions of an executable by the execution stage, and to order the starting or the stopping of an execution in the execution stage.

[0054] Security-related assumptions: At every moment it is defined which executable occupies the execution stage, if any, and portions of no other executable may be brought from the executables repository to the execution stage.

[0055] Control interaction between the APPARATUS FOR AUXILIARY COMPUTATIONS and the EXECUTION STAGE:

Role: To determine the operation performed by the apparatus for auxiliary computations.

Preferred Detailed Embodiments of the Formal Architecture

[0056] Fig. 3 is a prior art architecture of a securable public key protected device with a closed operating environment, 900, typically found in smart cards. These devices are single chip cryptocomputers, capable of performing symmetric and asymmetric

cryptographic functions, typically, generation of unique secret and public keys, encrypting, decrypting, authentication, and controlled access.

[0057] Such an architecture is satisfactory, if the operating system is secured and applications for use are written for host computers. The unit is a standalone safe keeper of application specific files which are addressable only by PKI authorized hosts. Each file is typically addressable only by an authorized host, under conditions specified by the host's PKI certificate. Typically the memory map is departmentalized, such that a host application programmer has no direct access to manufacturers' test code; the cryptographic library containing programs operable to perform sensitive cryptographic functions; or application specific data. Such a programmer is prevented from composing host programs to read out secret keys, and only authorized hosts may negotiate with electronic value purses or have access to confidential data files.

[0058] Security logic, 990, is operable to accelerate cryptographic functions, to allay external tampering, hacking, analysis of electronic data by operating the integrated circuit at out of scope operating conditions, and is operable to prevent an inimical entity access to parts of the memories, 910, 915, 920, internal busses, 970, or other peripherals. Control and Test Registers, 940, regulate and audit functionality of operation of peripherals and internal busses. The Watch Dog, 930, senses a nonfunctional central processing unit, CPU, 980. The Smart Card and Terminal interfaces, 995 and 998, enable communication with external terminals and readers. Random Sequence generators, 950, are operative to supply challenges to communicating devices, operable to assure that valid communication sequences are not recorded and reused by inimical devices, and to enable the cryptocomputer to generate unique secrets and secret keys. The Modular Arithmetic Processor, 960, is typically operable to execute popular public key algorithms, necessary for validation of certificates, generation and validation of electronic signatures, and for safe exchange of symmetric encryption keys.

[0059] This architecture is provably insecure if application programmers are allowed to write application programs in the operating system in 910.

[0060] Fig. 4. is a block diagram of a preferred embodiment of a double computing processor Public Key protected Trusted Computing Environment, TCE, 10, comprised of a security application module, SAM, 15, with an appended Trusted Application

Computing Environment, TACE, 25. The TCE, 10, communicates with a Host computing device, 220, operative to activate the TCE, and to enable authorized usage of the TCE. A plurality of Host Application SAMs are typically operable to authorize downloading of application specific executable programs and data; to negotiate transactions; and to store and recover data into and from the TCE, through the input output interface, 210. The CPU, 120, is preferably a reduced instruction set computer, with peripherals similar to the prior art security processor of Fig. 3. Typical components of the TCE SAM, are the modular arithmetic processor, 130, operable to process asymmetric cryptographic methods such as RSA, elliptic curve encryption and decryption, and digital signature algorithms, necessary for authenticating cryptographic certificates, application devices, execution of value transactions, and controlling processing of applications such that authorized applications are processed singly, without allowing interaction with other applications, and invasive procedures from the host or the application environment. Cryptographic peripheral, 50, performs symmetric encryption procedures, typically used in communication between remote host devices, and for storing data in the mass storage memory, 20. Memory device, 20, typically is not part of the monolithic circuit 10, and typically stores encrypted data, precluding inimical probing of connecting electronic signals between 10 and 20. Memory Controller 100, typically converts data, which is transmitted and received from 20 in large strings, similar to sectors of data read from hard disks and floppy disks. Sector accessed mass storage components are typically page readable and block erasable NAND Flash memory. System RAMs 140 and 160 are operative to maintain program functionality of the dual CPU units. The SHA1 Hash peripheral, 60, performs a one way function on long strings of data, operative to compress long strings of data to a 160 bit of data string, for public key signatures, executed in 130. The ROM peripheral, 70, is operative to contain all operational programs in the TCE SAM, including drivers for the cryptographic devices, drivers to store data in the electrically erasable and programmable read only memory, 40 and in the mass storage memory 20. The TCE SAM has its own data bus, S DATA, and address bus, which is not depicted. A plurality of control signals, each designated SCTRL, are operative to switch devices to limit the access of the application environment, 25, to specifically defined programs and data. Random Access Partitioned Revolving Single Application Memory, 80, is the platform

for executable TACE application programs. Typically, the TCE SAM transfers application programs into one section of 80, whilst the application environment typically executes application specific programs in parallel from a second alternate section of 80. Both prior to initiating an application session, and following an application session, typically, the TCE SAM Resets are operative to erase all residual data in volatile memories 80 and 160. In a typical application session the TCE SAM and the TACE application environment operate in parallel. Interactive communication between the two computing components is through the Security I/O Buffer, 150, typically using only predefined security commands, and access control to memory storage in the SAM and mass storage device for both reading and writing data, and controlling SAM negotiation functions. The Application RISC Processor, 190, typically processes an initial boot executed from ROM, 200, on the application environment in preparation for executing a SAM transmitted application session, and subsequently processes the authorized session typically written in a higher level language, typically interpreted in whole or in part by the interface program in 200. The Watch Dog, 280 is typically operative to sense a faulty sequence of commands in the TCE SAM, signifying that the TCE SAM has lost control of an application session. Typically, after sensing that the TCE SAM ceases to address the Watch Dog after a defined number of CPU clock cycles, 280 is operative to reset the TCE, 10.

[0061] Fig. 5 is a block diagram of a preferred embodiment of a double computing processor Public Key protected Trusted Computing Environment, TCE, 10, comprised of a security application module, SAM, 15, with an appended Trusted Application Computing Environment, TACE, 25. The TCE, 10, communicates with a Host computing device, 220, operative to activate the TCE, and to enable authorized usage of the TCE. The preferred embodiment of Fig. 5 is differentiated from Fig. 4 in that the mass storage device of Fig. 5, 320, is operable to be executable read only memory, freely accessed randomly. The CPU, 190, typically a RISC processor, is operable to execute programs directly from such memory, precluding the necessity of downloading programs into executable RAM memory, as in 80 of Fig. 4. As in Fig. 4, access to application memory typically is limited to a specific PKI protected application, whole or in part. In this preferred embodiment, Firewall Read Vector, 340, is typically prepared by the TCE SAM, programmed to enable only those parts of the specifically authorized

application prescribed by the Application Certification Authority in the Host's application certificate. The TACE, 25, is enabled to read and execute commands from the ROM memory, 320, subsequent to the TCE SAM switching the Address Mux, 330, to accept JADDR address signals.

[0062] A plurality of Host Application SAMs are typically operable to authorize downloading of application specific executable programs and data; to negotiate transactions; and to store and recover data into and from the TCE, through the input output interface, 210. The CPU, 420, is preferably a reduced instruction set computer, with peripherals similar to the prior art security processor of Fig. 3. Typical components of the TCE SAM, are the modular arithmetic processor, 130, operable to process asymmetric cryptographic methods such as RSA, elliptic curve encryption and decryption, and digital signature algorithms, necessary for authenticating cryptographic certificates, application devices, execution of value transactions, and controlling processing of applications such that authorized applications are processed singly, without allowing interaction with other applications, and invasive procedures from the host or the application environment. Cryptographic peripheral, 50, performs symmetric encryption procedures, typically used in communication between remote host devices.

[0063] Executable mass storage component, 320 is typically byte or word readable and block erasable NOR Flash memory. System RAMs 140 and 160 are operative to maintain program functionality of the dual CPU units. The SHA1 Hash peripheral, 60, performs a one way function on long strings of data, operative to compress long strings of data to a 160 bit of data string, for public key signatures, executed in 130. The ROM peripheral, 70, is operative to contain all operational programs in the TCE SAM, including drivers for the cryptographic devices, drivers to store data in the electrically erasable and programmable read only memory 40, and in the mass storage executable memory, 320. The TCE SAM has its own data bus, S DATA, and address bus, which is not depicted. A plurality of control signals, each designated SCTRL, are operative to switch devices to limit the access of the application environment, 25, to specifically defined programs and data.

[0064] Both prior to initiating an application session, and following an application session, typically, the TCE SAM Resets are operative to erase all residual data in volatile memory 160. In a typical application session the TCE SAM and the TACE

application environment operate in parallel. Interactive communication between the two computing components is through the Security I/O Buffer, 150, typically using only predefined security commands, and access control to memory storage in the SAM and mass storage device for both reading and writing data, and controlling SAM negotiation functions.

[0065] The Application RISC Processor, 190, typically processes an initial boot executed from ROM, 200, on the application environment in preparation for executing a SAM transmitted application session, and subsequently processes the authorized session typically written in a higher level language, typically interpreted in whole or in part by the interface program in 200. The Watch Dog, 280 is typically operative to sense a faulty sequence of commands in the TCE SAM, signifying that the TCE SAM has lost control of an application session. Typically, after sensing that the TCE SAM ceases to address the Watch Dog after a defined number of CPU clock cycles, 280 is operative to reset the TCE, 10.

[0066] Fig. 6 presents a preferred embodiment of a file system in the Trusted Computing Environment's (TCE) EEPROM, 40. The EEPROM is structured into a plurality of Directories. Each application is a directory of typically contiguous files.

[0067] Files are defined by an entry in the File Definition Table (FDT), 1650. The entry describes the file name, type, address, file size and the working mode. Elementary Files contain data for read and write or a program for a run file for application execution, or an Application Definition File (ADF), 1660. The ADF, which includes a File Allocation Table (FAT), defines an application structure.

[0068] Typically, a Public Key system is regulated by a Certification Authority or an agent of the Certification Authority (CA) who's Public Key is typically programmed into the main application of the TCE for certificate validation.

[0069] Applications (new directories) are programmed into the TCE. Each application has its own Application Definition File (ADF), 1660. Each application is owned by an application issuer and contains the files belonging to the application.

[0070] The non-volatile read-write memory of the TCE's SAM, 1670, typically has three separated logic partitions. The first secured partition, 1640, consists of the TCE SAM's initialization and personalization data comprising of the TCE SAM's certificate, which is signed by the Certification Authority; the TCE SAM's keys; an optional PIN to

initially activate the TCE's SAM; and Option bytes, which are operabale to limit cryptographic functions, as prescribed by governments.

[0071] The second partition, 1670, is intended for the File Definition Tables (FDTs), 1650. FDTs which point to Elementary Files (EF), e.g., 1610, 1630, or Application Definition Tables (ADFs), e.g., 1660.

[0072] The EFs may reside in the third logic partition, 1680 of the EEPROM, 40, or in the Memory Storage Extension (MSE), 20 or 320. Frequently updated Efs are typically reside in the TCE SAM's EEPROM, 40. ADF typically reside in the third logic partition, 1680. Typically, the data and executable files, for each application are defined in the EEPROM of the TCE, 40.

[0073] Fig. 7 is a preferred embodiment of a control vector mechanism operable to regulate access to executable NOR type Flash memory. A typical Application Definition File 1660 of Fig. 6, typically defines the bounds of an application with pointers in the TCE SAM EEPROM which relate to the bounds of the licensed application, or an authorized part thereof. Typically, the TCE SAM assembles and loads the Firewall Select Vector typically as defined in the Application Definition File.

[0074] In a preferred embodiment, the Firewall Select Vector is serially fed into a serial-in/parallel-out shift register, cells of which are depicted in 700, 710, 720, 730, and 740. The shift register is interleaved into the address decoder of the Random Access Application Flash ROM, 320 of Figs. 5 and 11. In this preferred embodiment, the memory is parsed into blocks. Typically, a block is enabled to be read when the decoded Block Select of the ROM Flash, 320, points to a block with a "1", AND the output of a cell of the shift register is a "1", 550. Typically, a "0" output of a shift register cell of the firewall select vector masks out a block of memory, thereby typically disabling access during an application session.

[0075] Typically, specific application blocks will contain only code and data relating to the same application. Blocks of memory in 690, typically depicted as 600, 610, 620, 630, and 640, are typically of equal size.

[0076] Fig. 8 is a simplified flow chart describing the Authentication Process & File Access procedure. Typically, in a PKI authentication process two members of an application are operable to communicate after mutual positive PKI identification.

[0077] At the conclusion of the process, the TCE SAM and Host SAM have negotiated their common application, have established file access permission, and have typically generated a DES session key for confidential data exchange.

[0078] The Certificate Exchange Process, 800 proves the existence of entities in an application. Validation of the host's SAM certificate proves to the TCE's SAM that there is a host authorized by the CA, whose public key is used to validate the Host's SAM and status. Public Key Verification, 810 typically proves the identity of the Host's SAM and TCE's SAM in preparation for executing an authorized application.

[0079] Access Rights to an Application, 820 for Read and Write Security Levels are typically defined in the File Definition Table (FDT), 1650. Typically, the Security Level defines which key(s) are required for a license to access files, wherein priorities are defined for reading and writing. Typically, the host sends commands to the TCE's SAM, operable to execute application in the application environment, 830. The TCE's SAM allocates application memory partition to the TACE.

[0080] Prior to executing an application, all-volatile memory is typically reset, 840.

[0081] Fig. 9 is a generic simplified block diagram representation of a preferred embodiment of a multi-media synchronized application configuration of double computing processor Public Key protected Trusted Computing Environment, TCE, 10, comprised of a security application module, SAM, 15, with an appended Trusted Application Computing Environment, TACE, 25. The TCE, 10, communicates with a Host computing device, 220, operative to activate the TCE, and to enable authorized usage of the TCE operable subsequently to input raw digital data preferably in clear text to be processed in the TACE environment. The Host data stream would typically be unenhanced purchased image data. Simultaneously, the TCE SAM typically decrypts purchased entertainment audio data interleaved with image enhancement parameter data from the Mass Storage Memory, 20, operable to synchronize audio data with image data in the application environment, 25.

[0082] In a preferred embodiment the encrypted data stream from the mass storage memory, 20, would typically include a dubbed in sound track of the purchased movie wherein the unenhanced image data is typically recorded on CDROM disks.

[0083] The analog audio outputs are typically output on stereo D/As, 270, and the enhanced synchronized images are output on 170.

[0084] An attacker is able to copy the analog data on outputs 170 and 270, but is typically unable to retrieve quality digital data.

[0085] Fig. 10 is a block diagram of a preferred embodiment of a double computing processor Public Key protected Trusted Computing Environment, TCE, 10, comprised of a security application module, SAM, 15, with an appended Trusted Application Computing Environment, TACE, 25. The TCE, 10, communicates with a Host computing device, 220, operative to activate the TCE, and to enable authorized usage of the TCE. An additional A/D converter operable to assemble an image is integrated into the TACE, wherein the image is typically stored in the local TACE RAM.

[0086] The configuration with a A/D input is typically operable to execute fingerprint identification. The TACE typically enhances the input image wherein blemishes, unconnected lines and curves are repaired. Features are typically extracted from the fingerprint image and the proximity of the measured image is compared with a stored feature template in the application memory to statistically authenticate personal identification.